

# DL Assignment 4

Kaustubh Hiware, 14CS30011

Mar 7, 2018

- 30 iterations / epochs for training.
- For our neural network, number of input nodes were 784 (since each image was 28\*28 pixels.). Number of hidden nodes was a variable. Number of output nodes will be number of classes, that is 10.
- A seed of 123 was chosen randomly, to ensure result reproducibility.
- The learning rate was set to 0.005, for a smoother convergence.
- Batch size chosen as 100, to accommodate individuality of samples, and optimise number of batches.
- Possible to restrict iterations via --iter flag.

Script for running all combinations: run.sh, total output: output.txt

## Train accuracy

Hidden unit	LSTM	GRU
32	71.69833%	71.19%
64	79.083336%	74.598335%
128	80.42167%	45.818336%
256	79.34%	10.0%

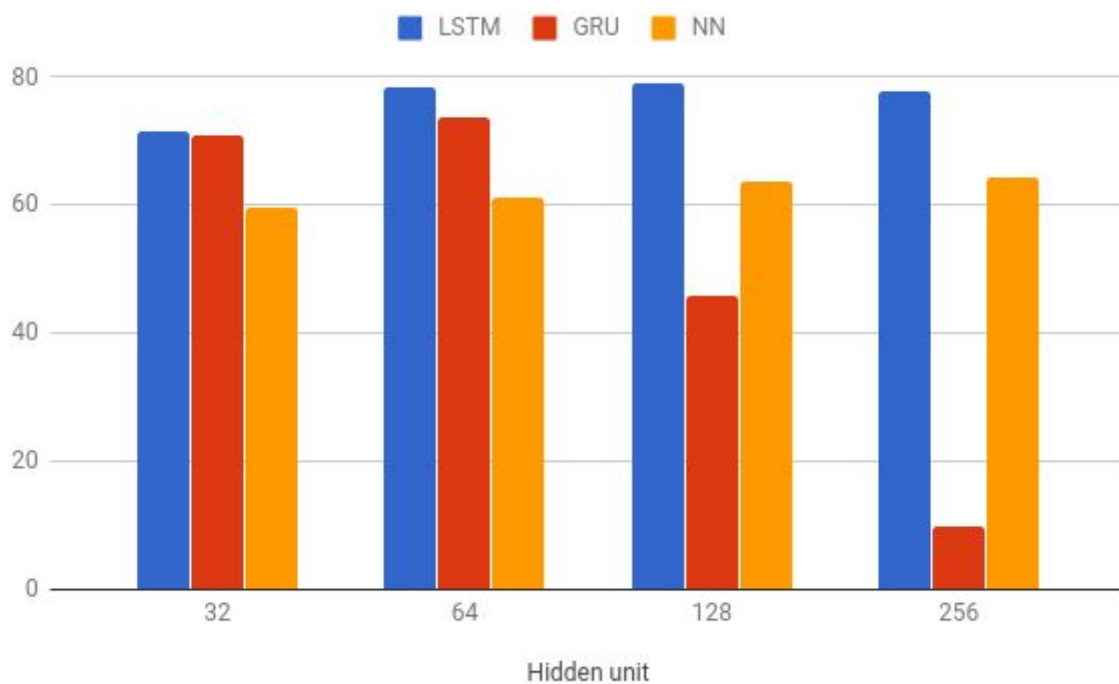
## Test accuracy

Hidden unit	LSTM	GRU
32	71.590004%	70.75%
64	78.26%	73.76%
128	79.149994%	45.829998%
256	77.69%	10.0%

## Accuracy comparison plot (Test)

The neural net for previous assignment corresponds to NN. Note that the previous assignment had 50 iterations, and a fixed hidden unit. Thus, after running NN with the required parameters, we have:

Hidden unit	Training	Test
32	62.35%	59.72%
64	62.89%	61.21%
128	65.51%	63.83%
256	66.54%	64.37%



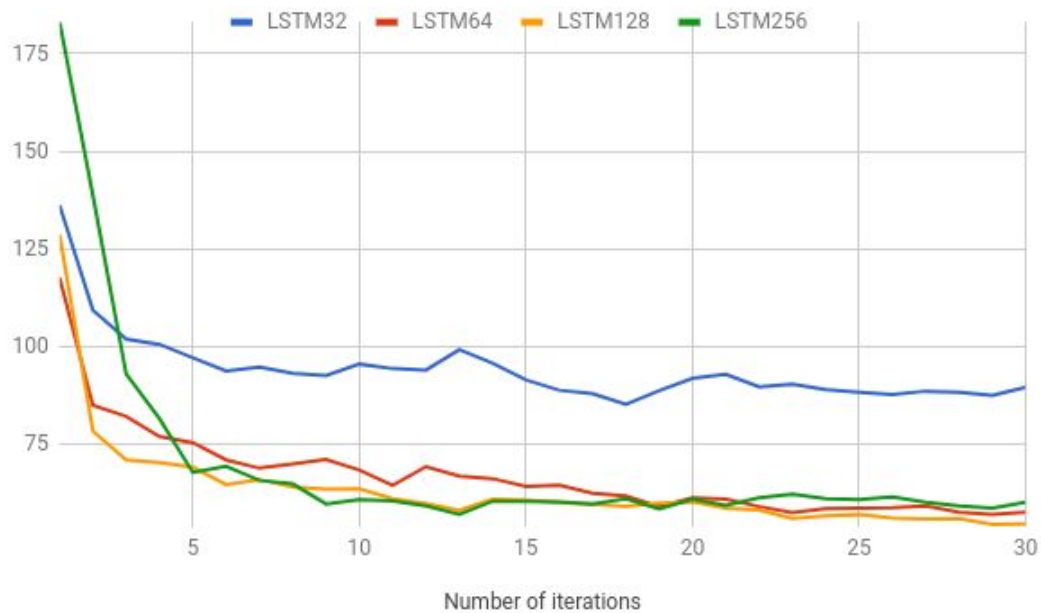
In terms of performance, we see LSTM > GRU and LSTM > NN.

For hidden units 32 and 64, LSTM and GRU have comparable performances, outperforming NN by a significant margin.

However, for hidden units 128 and 256, the size causes an increase in cost in GRU, due to which accuracy is severely affected. Thus, in terms of performance, LSTM > NN > GRU.

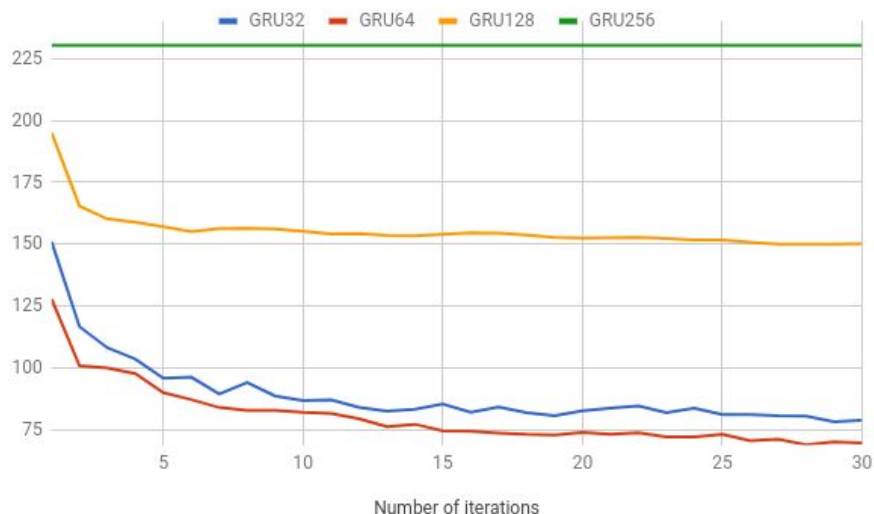
## Cost reduction per iteration

## LSTM



We see for LSTM32 (in blue), the minimum cost reached is much higher, and for higher hidden\_units, the cost /loss all seem to coincide.

## GRU



For GRU32 and GRU64, the final cost / loss seems to coincide. However, with an increase in hidden layer size, the cost increases drastically. In GRU128, we see an initial drop in cost, but it never reaches the final cost of GRU32. However, in case of GRU256, the cost does not seem to decrease at all (refer output.txt) (changes in order of 0.00001). This explains the poor accuracy in case of GRU256.

## Sample output:

```
$ python train.py --train --hidden_unit 32 --model lstm --iter 5

> hidden unit flag has set value 32
> model flag has set value lstm
> Now Training
> Running for 5 iterations
> Hidden size unit 32
2018-03-07 13:26:26.509205: I
tensorflow/core/platform/cpu_feature_guard.cc:137] Your CPU
supports instructions that this TensorFlow binary was not compiled
to use: SSE4.1 SSE4.2 AVX AVX2 FMA
Iteration 0, Cost: 136.257942746, Accuracy: 58.9999961853
Iteration 1, Cost: 109.432292188, Accuracy: 75.0
Iteration 2, Cost: 102.023282509, Accuracy: 58.9999961853
Iteration 3, Cost: 100.68580843, Accuracy: 66.0
Iteration 4, Cost: 97.252896684, Accuracy: 60.0000038147
Parameters have been trained and saved!
Train Accuracy: 70.10333 Test Accuracy: 70.420006
$                               1.7m  Wednesday 07 March 2018 01:28:07 PM IST
```